

Enhanced Distance Based Broadcasting Protocol with Reduced Energy Consumption

Patricia Ruiz
University of Luxembourg,
Luxembourg
patricia.ruiz@uni.lu

Pascal Bouvry
University of Luxembourg,
Luxembourg
pascal.bouvry@uni.lu

ABSTRACT

One important challenge for broadcasting in mobile ad hoc networks is the well known broadcast storm problem. Most of existing techniques to prevent it usually lies in reducing the number of forwarding nodes in the process. One of these algorithms (from the state of the art) is the distance based broadcasting algorithm, that tries to minimize this number considering only as candidates of forwarding those nodes far from the source. In this paper, we propose a cross layer design for enhancing the distance based broadcasting protocol in terms of energy consumption. Instead of using the distance, we are considering the reception signal strength. That is more realistic since the theoretical transmission range is not accurate in environments with obstacles. The necessary transmission power to reach an intended device is obtained using the beacons. If the furthest node can be reached using less power than the default value, the transmission power is reduced and thus, we save energy. Different proposals for enhancing the algorithm are proposed, and they not only save energy but also highly reduce the number of collisions.

KEYWORDS: Energy efficiency, mobile ad hoc networks, distributed system, cross layer design, distance based broadcasting.

1. INTRODUCTION

Due to the extraordinary amount of personal wireless devices that already exist on the market, the idea of

being able to communicate using only these smart devices is attracting the attention of many researchers. This kind of networks is called mobile ad hoc networks, or MANETs, and is composed of a set of wireless devices able to communicate between them. The main feature of MANETs is that they do not need any previous existing infrastructure that would help for routing the packets from a source node to a destination. These networks have many challenging aspects like the appearance and disappearance of devices due to the limited transmission range, the battery life, the mobility of devices, obstacles in the environment, the network partitioning, etc. Many works have been done trying to overcome all these undesirable behaviors.

The intrinsic broadcast nature of wireless networks makes broadcasting process one of the most suitable algorithms for neighbor discovery, routing, etc. Moreover, broadcasting is considered as one of the basis for many high level applications. Even other protocols assume the existence of a broadcast service. That is the reason why many researchers try to optimize these algorithms by maximizing the number of nodes reached and minimizing the use of the required network and device resources [1].

One of the main problems in dissemination is the broadcast storm problem [2]. Not only the problem was presented in [2] but some different techniques for minimizing its effects were introduced, like (1) the probabilistic scheme where nodes resend the message with a predefined probability. (2) The counter based approach that forwards the message in terms of the number of copies received. (3) The distance based

technique that considers candidate nodes for forwarding those further from the source than a predefined distance. (4) In the location based approach, the receiver knows the position of the source, so it is able to calculate precisely the additional area covered with the forwarding. If it is less than a predefined value, the message is not resent. (5) Finally, in the cluster based scheme, nodes are distributed in clusters. Only those nodes considered as head or gateway are candidates for forwarding. Those candidates will use one of the previous techniques to determine whether to rebroadcast or not. All these approaches try to minimize the number of forwarding nodes. In this work, we are considering the distance based broadcasting protocol (DB hereinafter), that aims at selecting forwarding nodes in terms of the distance between the receiver and the source node, and we are enhancing it by minimizing the transmission power every node uses for the broadcasting process in order to save energy and reduce the number of collisions.

We are interesting in the energy consumption because ad hoc networks are composed of devices that rely on batteries. Reducing the transmission power will directly increase the battery life of the nodes and thus, the network lifetime. This topic has been extensively addressed in the literature [3–7].

The contributions of this paper are three folds: (1) adding energy efficiency features to the distance based approach by reducing the transmission power of the source nodes, (2) analyzing the influence that reducing the transmission power has over other nodes in terms of the number of collisions or the interference level, and (3) studying the behavior of the algorithm according to the setting of the delay.

The remainder of this paper is organized as follows. Next section presents a small state of the art in the topic. Section presents the original distance based broadcasting algorithm, and the improvements added to it are explained in Section . The parameters used for the simulation and the results obtained are shown in Section and Section , respectively. Finally, Section concludes the paper.

2. RELATED WORK

In this work, we are considering an algorithm of the state of the art: the distance based approach. It is

possible to find different proposals in the literature, just like [8], where authors designed a distance based broadcasting protocol that does not exchange any *hello* message (to alert devices about the presence of neighbors in range). A counter based scheme is included in this approach, so that the distance threshold for re-sending the message varies in terms of the number of copies heard of the message. In [9], an improved version of DB is presented combining the advantages of distance-based and counter-based broadcast schemes, and considering also the remainder energy of sensor node.

As we mentioned before, the energy consumption in mobile ad hoc networks is a hot topic, since devices can run out of battery provoking the network degradation. We mention below some of the solutions that have been proposed for saving energy while disseminating in ad hoc networks. Gomez et al. showed in [4] that a variable transmission range can outperform a common range transmission approach in terms of power saving, increasing the capacity at the same time. In [10], nodes exchange information in the beacons in order to know the transmission power needed to reach the two hop neighbors. The source node examines if it is worth excluding the furthest node from the one hop neighborhood and reduce the transmission range to reach the new furthest neighbor. An approach to estimate the local density using an analytical model is used in [3] to set the transmission range according to this estimation. Studies on energy efficient algorithms for finding the minimum-energy broadcast tree (MEBT) have been proposed [5, 6], and also in [7] a shared multicast tree built in a distributed fashion with minimum energy is presented, where the transmission power is either fixed or adjustable.

3. DISTANCE BASED BROADCASTING ALGORITHM, DB

As we explained before, DB is one of the different schemes proposed for minimizing the effects of the broadcast storm problem when disseminating information in wireless networks [2]. The protocol makes use of the distance between the source node and the receiver (the procedure to obtain this metric is not specified in [2]). The idea is that a node receiving a broadcast message for the first time will compute the distance to the source node. If this distance is small, the

contribution to the dissemination performing this forwarding is negligible (see Figure 1), and therefore, the message is not resent.

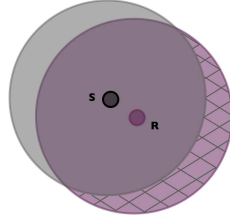


Figure 1 Additional Coverage Performed.

As stated in [2], the distance from source node to the receiver is clearly related to the additional coverage obtained in case of forwarding, so it can be used as a metric. Only nodes that are separated at least a minimum distance from the source node resend the message. This minimum distance is a predefined threshold, D . The protocol also includes a delay before forwarding a received message, and if the same message is heard more than once (during this waiting time), the delay is cancelled. The distance between the current node and the new sender is calculated and if it is higher than D , the message is resent.

Figure 2 represents the functioning of the algorithm. Considering node A broadcasts a message m , nodes B and C will not resend m because the distance from those nodes to A is smaller than D . Nodes E , F and G will wait for a random number of slots. If node F finishes the waiting time first, it will forward the message and, thus, node E will hear it and calculate the distance from node F . As the distance is smaller than D , node E will drop the packet. The pseudocode of the protocol is shown in Algorithm 1.

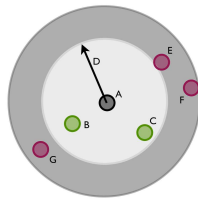


Figure 2 Mechanism of DB.

4. ENHANCED DB

In this section we explain the procedure followed to implement the broadcasting algorithm, and also the

Algorithm 1 Pseudocode of DB.

Data: m : the incoming broadcast message.
Data: r : the node receiving broadcast message.
Data: s : the node that sent m .
Data: d : the distance between r and s .
Data: $dmin$: the minimum distance between r and any s .
Data: D : the distance threshold.

```

1: if  $m$  is received for the first time then
2:   calculate  $d$ ;
3:   update  $dmin$ ;
4:   if  $dmin < D$  then
5:      $r \rightarrow$  drop message  $m$ ;
6:   else
7:     waiting = true;
8:     wait random number of slots;
9:     goto 23;
10:  end if
11: else if waiting then
12:   calculate  $d$ ;
13:   resume waiting;
14:   waiting = false;
15:   if  $d < dmin$  then
16:     update  $dmin$ ;
17:   end if
18:   if  $dmin < D$  then
19:      $r \rightarrow$  drop message  $m$ ;
20:   else
21:     goto 23;
22:   end if
23: end if
24: transmit  $m$ 

```

improvements introduced to the original protocol, DB.

4.1. Implementation

For calculating the distance between a source and a destination, the most common technique is either assuming a GPS service or considering the signal strength. In this work, we do not assume that all devices in our network must provide a GPS service. Therefore, we use the signal strength of the received packets to estimate how far two nodes are.

There are some approaches where the distance between the nodes is estimated according to a selected propagation path loss model for the channel. In [9], authors consider free space propagation model, and the distance is obtained from that determined equation using the reception power (assuming all devices use the same transmission power).

In this work, we are not assuming any predefined propagation path loss model. The received power is related to the distance, although we are not interested in the distance itself but in the energy lost during the transmission. Considering this, a device close to the source node but with any obstacle weakening the signal, will

forward the message contributing to the process in an area where the dissemination of the message is arduous. Using this implementation we are also aware of the non perfect shape of the transmission range as a disk, since it depends on the environment.

In our implementation, the threshold D is not in terms of distance (m) but power (dBm). We called it *borders_Threshold* as it defines the nodes that are considered to be far from the source and therefore close to the border. The value used for this parameter is *-90 dBm*. This value was experimentally chosen, and any value below it supposes that the source and destination nodes are separated at least 2/3 of the maximum coverage. A node is not able to decode a received packet if the reception power is lower than *-95 dBm*, this is called the *end_Threshold*. Therefore, all nodes whose reception energy vary from *[-95, -90]* dBm are candidates of forwarding the broadcasting message.

Every device sends a *hello* message (or beacon) to alert devices within a close area about their presence. A device receiving these beacons is able to keep track of all neighbors around. We assume that all devices send the beacons with the same transmission power.

We are considering here a cross layer design where the physical layer informs the upper layers about the received signal strength of each beacon. In this situation the algorithm is able to take decisions depending on this value. When a broadcast message is sent, the receiving node will check the reception power, if it is below the *borders_Threshold* (*-90 dBm*), it will consider itself as a bordering node and thus, sets the delay.

4.2. Enhancements

As we are dealing with ad hoc networks, and devices depend on battery, saving energy supposes one critical aspect. One of the new features added to DB is reducing this energy consumption. For improving DB, we also consider different settings the protocol establishes for the delay when a message is received, and we analyze the behavior of the algorithm.

4.2.1. Reducing Transmission Power: In any wireless transmission, as the electromagnetic wave propagates through the space, the power of the signal suffers from path loss attenuation causing a reduction in the signal power. The relation between the transmitted power

and the power finally received at the destination directly depends on the loss suffered during the transmission. Equation 1 represents the relation in terms of dB.

$$receivedPower = transmittedPower - loss \quad (1)$$

We assume that all nodes send the *hello message* with the same transmission power (*16.02 dBm*). Thus, a node receiving a beacon will be able to estimate the loss that packet suffered during the transmission, using the reception power detected at the physical layer.

Every node keeps and updates the reception power of each of its neighbors in a list. When a device wants to send a broadcast message, it will be able to estimate the loss the packet will suffer (as we assume a packet traversing in a direction will experiment the same loss as another one traversing in the opposite direction). If

a node can estimate the loss the packet is going to suffer, it will be able to reduce its transmission power and use only the necessary one to get the furthest one hop neighbor. Thus, reducing the transmission power for sending broadcast messages directly decreases the energy consumption of the device, without degrading the performance of the broadcasting process as we do not consider losing the connection with any neighbor.

We are reducing the transmission power, so that the furthest node is receiving the packet with the minimum reception power allowed to correctly decode the message. That means, its reception power should be the *end_Threshold*. When the loss the packet suffered due to the propagation is calculated, the node can estimate the transmission power needed to reach the furthest neighbor in the one hop neighborhood. The new reduced transmission power can be calculated as shown in Equation 2.

$$transmissionPower = loss + endThreshold \quad (2)$$

Once the new transmission power is estimated in terms of the reception energy of the beacons, it is necessary to consider that the devices do move and the information is not trully precise as *hello messages* are sent every 1 second. Therefore, we are considering a margin of error (*margin_Forwarding*) and the value selected

was 0.5 dBm. This value was experimentally chosen, and it was estimated to compensate the possible movements of the node since last beacon reception.

From Equation 2 it is possible to estimate the maximum transmission power needed to reach the furthest neighbor in the one hop neighborhood. If it is less than the default transmission power, we reduce it in order to save energy, as the extra energy used is useless (see Figure 3). Therefore, reducing the transmission range from r to r' decreases the energy consumption with no detriment of the network connectivity.

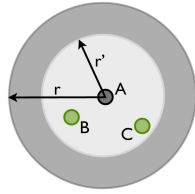


Figure 3 Reducing the Transmission Power of Nodes.

Reducing the transmission power for sending broadcast messages not only improves the energy consumption in wireless networks, but also reduces the interference level of devices in a close area. We know that each device has the *end_Threshold* from which on, if the received signal strength is lower, the device will not be able to recover the data transmitted, but this reception will be considered as noise and will increase the interference level of the device [11].

We can observe in Figure 4a that both nodes *A* and *B* use the default transmission range r . In this situation node *C* is not in range either with node *A* or *B* but it suffers from their interference. However, if both nodes *A* and *B* reduce their transmission range to the maximum needed r' , as it is shown in Figure 4b, node *C* will not receive anything from *A* or *B*.

As it was explained in [12], the transmission power affects many aspects of the network just as the transmission range, the connectivity of the network (the lower the power, the smaller the transmission range), the performance of the medium access (it depends on the number of nodes within range), the capacity of the network, etc. The network connectivity in this case is not decreased, since nodes try to reduce the transmission power but considering all neighbors in range. Nei-

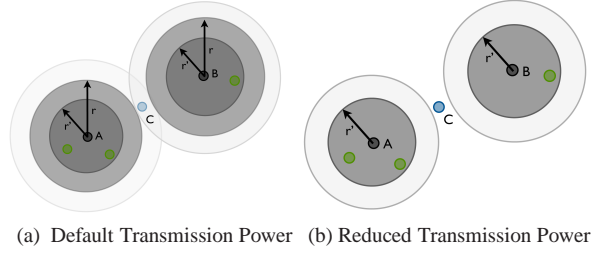


Figure 4 Interference and Trans. Power Relation.

ther the contention for the medium access is, exactly for the same reason explained before. But the capacity is increased as the transmit power level is reduced by decreasing the interference area (it is proportional to the square of the transmission range).

4.2.2. Using Different Delay Techniques: DB stops the random delay when a repeated message is heard. Then, if the distance from the new source node is smaller than the threshold *D*, the message is discarded and no retransmission is performed. Otherwise, the forwarding starts.

Instead of stopping the delay when a repeated message is heard, we consider the possibility of keeping track of the received energy and continue the delay. Once it is finished, the forwarding decision is taken according to maximum received power.

In DB the delay is randomly chosen from a predefined interval. We propose to adopt a similar scheme as the one used in [13, 14], where the delay is fixed and it is proportional to the distance from the receiver to the source node. In this situation, a node closer to the source will rebroadcast later than a node far from it.

In our work, we are studying the behavior of two different techniques and we are comparing them to the original proposal of DB:

1. In the first one a fixed delay inversely proportional to the received power is considered. In Equation 3, the procedure to calculate the delay in terms of the reception power is shown.

$$powerDelay = \frac{-1}{rxPower - borders_Threshold - 1} \quad (3)$$

Algorithm 2 Pseudocode of enhanced DB.

Data: m : the incoming broadcast message.
Data: r : the node receiving broadcast message.
Data: s : the node that sent m .
Data: p : the received signal strength of m sent by s .
Data: $pmin$: the minimum signal strength received for m from any s .
Data: $borders_Threshold$: the signal strength threshold.

```
1: if  $m$  is received for the first time then
2:   calculate  $p$ ;
3:   update  $pmin$ ;
4:   if  $pmin > borders\_Threshold$  then
5:      $r \rightarrow$  drop message  $m$ ;
6:   else
7:     waiting = true;
8:     wait time rand/randDist/fixedDist;
9:     goto 20;
10:  end if
11: else if waiting then
12:   calculate  $p$ ;
13:   if  $p > pmin$  then
14:     update  $pmin$ ;
15:   end if
16: end if
17: if  $pmin > borders\_Threshold$  then
18:    $r \rightarrow$  drop message  $m$ ;
19: else
20:   transmit  $m$ ;
21: end if
22: waiting = false;
```

If a node is setting a delay, it means, the node is a *border* node, otherwise the node is not considered as a candidate to forward the message, and therefore, no delay is set. All *border* nodes receive the message with a reception power that can vary between the *borders_Threshold* (-90 dBm) and the *end_Threshold* (-95 dBm). Therefore, considering Equation 3, we can check that the maximum possible delay is 1 second. The higher reception power (closer neighbors), the longer the delay, and vice versa.

2. The second proposal considers a random delay chosen from an interval whose size also varies with the reception power. That is, the waiting time will be chosen between $[0, powerDelay]$ calculated as in Equation 3. We can check that the delay varies from 0 to 1 second.

We are comparing different techniques: (1) the delay is chosen randomly from the interval $[0, 1]$ s; (2) the delay is fixed with the value *powerDelay*, and finally (3) the delay is chosen randomly from the interval $[0, powerDelay]$ s.

The pseudocode of the enhanced distance based broadcasting protocol proposed in this work is shown in Algorithm 2.

5. SIMULATIONS

In order to experimentally evaluate the performance of the new features we are adding to the distance based broadcasting protocol, we are using ns-3 simulator [15, 16]. ns-3 is a discrete event simulator written in C++ and with an optional Python scripting API. To validate the enhancements we are providing to the algorithm, we are comparing the original DB with the new one in terms of different parameters. These aspects are:

- The total energy saved when reducing the transmission power of the source and all the forwarding nodes.
- The coverage achieved by the broadcasting message.
- The energy used per forwarding message.
- The network usage, in terms of the number of nodes forwarding the broadcast message.
- The number of collisions due to the broadcasting process.

As we are dealing with mobile ad hoc networks, it is necessary to set a mobility model for the devices. In this case we are using the *random walk* also known as *brownian motion mobility model* [17], in which nodes move with a speed and direction randomly chosen during a fixed amount of time, in our case 20 seconds. If a node hits one of the boundaries of the area, it rebounds on the boundary with a reflexive angle and speed. The simulation environment used is a square area of 2000 m side (4Km²). The speed of the nodes can vary from 0 to 2m/s (between 0 and 7.2Km/h).

We measure our experiments with different network densities: the number of nodes varies from 100 up to 700 in steps of 100 devices. In Table 1, we present a summary with the parameterization we are using for the simulations.

As we explained before we are comparing different variants of the broadcasting distance based protocol (DB). In Table 2, all the different proposal are explained.

6. RESULTS

For obtaining reliable results we are considering 100 different independent topologies for each of the seven

Table 1 Parameterization Used

<i>Number of devices</i>	100-700
<i>Speed</i>	[0, 2] m/s
<i>Size of the area</i>	2000 m × 2000 m
<i>Transmission power</i>	16.02 dBm
<i>end.Threshold</i>	-95 dBm
<i>borders.Threshold</i>	-90 dBm
<i>margin.Forwarding</i>	0.5 dBm
<i>Delay interval</i>	[0, 1] s
<i>Direction change</i>	every 20 s

Table 2 Different Proposals

<i>S.Rad.DB</i>	original DB
<i>S.Rad.EE</i>	DB with energy estimation (EE)
<i>S.Fix.DB</i>	DB with fixed delay
<i>S.Fix.EE</i>	EE with fixed delay
<i>S.Pow.DB</i>	DB with delay $\in [0, \text{powerDelay}]$
<i>S.Pow.EE</i>	EE with delay $\in [0, \text{powerDelay}]$
<i>NS.Rad.DB</i>	DB not stop delay
<i>NS.Rad.EE</i>	EE not stop delay
<i>NS.Fix.DB</i>	DB with fixed delay & not stop delay
<i>NS.Fix.EE</i>	EE with fixed delay & not stop delay
<i>NS.Pow.DB</i>	DB delay $\in [0, \text{powerDelay}]$ & not stop delay
<i>NS.Pow.EE</i>	EE delay $\in [0, \text{powerDelay}]$ & not stop delay

densities we are using. We are considering very partitioned networks with 100 devices in $2000 \text{ m} \times 2000 \text{ m}$ (4 Km^2) and we are increasing the number of nodes by 100 in each considered scenario until we reach to 700 devices what means the density varies from 25 devices/ Km^2 , which is a really sparse network, to 175 devices/ Km^2 . In the experiment the network evolves for 30 seconds (so that the devices are uniformly distributed all over the simulation area) and at that moment a node sends a broadcast message. The simulation stops after 50 seconds.

Statistical analysis have been made to all the results presented in this paper. This study has been done using the *boxplot* function from *Matlab*. In the displayed *boxplots*, the bottom and top of the boxes represent the lower and upper quartiles of the data distribution, respectively, while the line between them is the median. The whiskers are the lowest datum still within 1.5 IQR of the lower quartile, and the highest datum still within 1.5 IQR of the upper quartile. The crosses are data not included between the whiskers. Finally, the notches in the boxes display the variability of the median between samples. If the notches of two boxes are not overlapped, then it means that there is statistical significant difference in the data with 95% confidence. Due to the lack of space not all the statistical results are presented, in case there is something remarkable, we mention it.

6.1. Coverage

In terms of the coverage achieved by the broadcasting process, we found that there are no statistical differences in the five first densities studied between any of the proposals. When dealing with 600 devices there is significant difference between NS_Fix_DB and NS_Rad_DB, where the former gets better result than the latter. For the densest network, NS_Pow_EE reaches more devices with statistical difference than NS_Pow_DB, NS_Rad_EE and S_Pow_EE. That means, there is no loss of coverage of the new proposed protocols regarding the original one (S_Rad_DB). This is a very important result since the robustness of the proposed solutions in terms of the coverage achieved is not decreased with statistically difference.

The coverage achieved by the broadcasting process is highly satisfactory with reasonable use of the network resources. For the 100, 200 and 300 densities, the network is very sparse and the coverage achieved is very small (the minimum for 100 devices is 2.99%), but as the density grows, the coverage is also increasing. In networks with 600 and 700 devices the minimum coverage achieved among all protocols is 94.375% and 97.16%, respectively, while the number of forwarding used is lower than 31%.

6.2. Energy

In Figure 5, the evolution of the energy saved during the broadcast process of each different proposed algorithm is shown. It is measured as the total amount of energy saved by the broadcasting process. It is possible to see how the energy savings are increasing up to a moment in which they start to decrease. This is due to the fact that in our implementation the protocol is reducing the transmission power only when there are no neighbors in the border of the transmission range, thus, it does not make the connectivity of the network worse. But when the network is very dense, the number of neighbors increases and also the possibility of finding any close to the limit of the transmission range, therefore no reduction in the transmission power is performed.

It is also shown in Figure 5 that, in denser networks, the different variants that stop the waiting time when the same message is heard twice save more energy

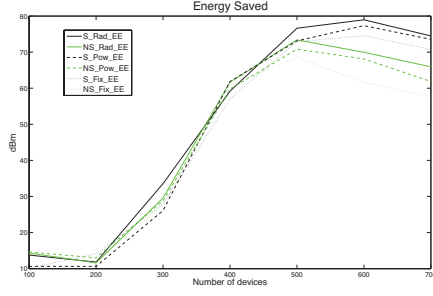


Figure 5 Energy Saved in the Broadcasting Process.

than the ones that let the delay finishes. This is logical, since nodes that keeps waiting until the end of the delay could detect a close neighbor resending the message and thus, it cancels its reemission. When a forwarding is cancelled the possible energy saved is not increased. Therefore, it is also necessary to check the number of forwardings that each protocol uses.

The average value of the number of forwardings performed by each proposal is not included in the paper because of the lack of space. Results show that for denser networks, as the number of devices increases, the number of forwarding nodes is higher for those protocols that stop the waiting time when the message is received for the second time. The statistical study shows that, in general, there are no significant differences in sparse networks. For 500 devices and up, all the protocols dealing with the energy consumption have significant difference with the ones with the same configuration but not reducing the energy power. This can be explained because when the transmission power is reduced, the interval where a node is candidate to resend the message is closer to the source node, and therefore, there exist more possible forwarding nodes. Statistical results for 600 devices also show that there are significant differences between protocols stopping or non stopping the delay. Figure 6 shows the *boxplot* obtained for the 700 devices network, where all the difference are clearly shown. In this case, apart from the difference in general between stopping the delay or not, NS_Fix_DB and NS_Pow_DB are the ones performing less forwardings with statistical differences with all the rest of proposals.

As we mentioned before, the algorithms saving more energy were the ones that performed more number of

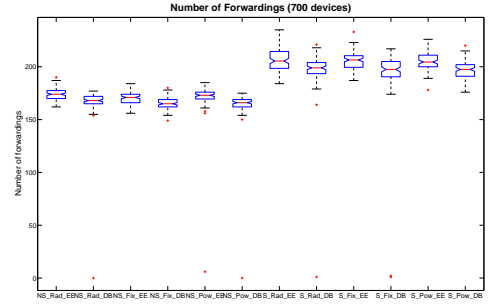


Figure 6 Number of Forwardings Performed.

forwarding in the broadcasting process, that is, the algorithms that stop the waiting time when the same message is received twice. For that reason, a measure of the energy saved in terms of the number of forwardings performed was needed. In Table 3, the average value of the energy consumed per forwarded message is shown. The first row represents all the different algorithms presented that do not consider energy efficiency, thus, they all consume the same power: 16.02 dBm. The algorithms consuming less energy per forwarding in each network density are marked in **bold font**.

We also made statistical test in this case, and the results show that there are significant differences between any proposed algorithm with energy considerations and the ones that do not reduce the transmission range (Any_DB), for all of the different network densities. In Table 3, the significant statistical differences between NS_Rad_EE and the rest of algorithms are shown, as it is the only approach that has significant differences with other proposals for different densities. The algorithms with the background in *light grey* color are statistically worse than NS_Rad_EE. The background in *dark grey* color means statistically better behavior. Between all the algorithms reducing the transmission range, no significant difference was found in networks with 100, 200 or 300 devices. For densities with 400 devices and up, statistical differences appear. As the network density increases, the number of algorithms statistically worse rises, and finally in the densest network NS_Rad_EE is statistically the best one compared to all the rest of algorithms. It is also remarkable that NS_Fix_EE and S_Fix_EE are the worst ones with significant differences.

Table 3 Average of Energy Used per Forwarding

Densities								Ranking
100	200	300	400	500	600	700		
Any_DB	16.02	16.02	16.02	16.02	16.02	16.02	16.02	
NS.Rad.EE	7.17±28.74	14.71±143.66	15.20±511.13	15.38±769.91	15.51±527.90	15.58±508.30	15.64±82.89	1
S.Rad.EE	10.36±41.92	14.54±122.45	15.21±578.53	15.41±802.23	15.52±630.89	15.59±441.65	15.66±165.73	2
NS.Fix.EE	8.63±28.66	14.59±113.88	15.27±510.09	15.44±633.96	15.53±595.56	15.63±419.95	15.68±090.33	5
S.Fix.EE	11.49±34.81	14.62±175.32	15.27±523.29	15.43±874.87	15.55±537.80	15.62±308.80	15.68±128.02	6
NS.Pow.EE	9.12±34.35	14.58±148.02	15.24±542.28	15.42±710.59	15.52±559.97	15.60±133.88	15.66±278.38	3
S.Pow.EE	11.09±34.40	14.62±140.88	15.17±476.94	15.42±700.50	15.52±756.06	15.60±427.71	15.66±134.36	4

We did a ranking between each proposal in every density numbering each one from 1 (the best) to 6 (the worst). Summing up the rank for each density we get an overall ranking for the proposal that is shown in the last column of Table 3. The proposal obtaining the best position in the ranking is NS_Rad_EE.

According to Table 3, the energy consumed by a node forwarding a message is 15.68 dBm in the worst case, and the original one (S_Rad_DB) is consuming 16.02 dBm. This is considering a logarithmic scale, so if we convert this value to watt, DB consumes 40 mW and the worst case of any proposal is 36.98 mW. That difference supposes a saving of 7.55% in each forwarding in the worst case of the densest network. For the sparser network where devices are able to reduce more the transmission power, the energy saved is higher. The worst case in the 100 devices network, consumes 11.49 dBm, which means 14.09 mW, what supposes 64.77% of saving. In the best case (NS_Rad_EE), the reduction supposes 86.97% per forwarding message.

6.3. Collisions

We consider a collision can be produced due to the reception of a packet when the device is already synchronized or transmitting. In Table 4, the number of collisions produced due only to the broadcasting process is shown for each density and algorithm. It is possible to check how in sparse densities, the number of collisions is low, i.e. with a 100 devices network the maximum average number of collisions is 0.14 for all the different protocols. It is possible to check the big difference between the two different versions of the protocols, the one that stops the waiting time and the one that does not, no matter which value of the delay is set. In all protocols that do not stop when a copy of the message is heard, the number of collisions is much lower than in the ones stopping. On one hand, if

we consider, for example, the original DB (S_Rad_EE) with 700 devices in the network, the average number of collisions is 1603.83. On the other hand, the same protocol that just finishes the delay (NS_Rad_EE) has an average number of collisions of 69.96, what means 95.63% less. This value is the maximum percentage in average of collisions reduction that can be found in the network, but from 400 devices and up, the minimum percentage of reduction is higher than 85%.

Statistical results regarding the collisions show that there is no significant difference in 100 devices. For the rest of densities there is significant difference between the non stopping approaches and the stopping ones. For networks with 400 devices up to 700, the NS_Rad_EE is the best one all over the rest (except NS_Rad_DB) with significant differences.

In Figure 7, the boxplot of the statistical results for the number of collisions in the network with 700 devices is shown. As it is possible to see, there is no statistical difference between any two protocols with the same configuration for the delay but considering the reduction of the transmission power or not (i.e. between NS_Rad_DB and NS_Rad_EE).

There are significant differences between two different configurations of the delay: it does not matter how the value for delay was estimated but if the node is stopping it and forwarding immediately the message arrives for the second time, the number of collisions is higher. That is because more than one node can hear the message for the second time, thus, they will stop the delay and resend the message at the same time provoking a collision. Considering the three different configurations for the delay when the node does not stop the waiting time, it is possible to see in the *boxplot* that there are statistical significant differences between all of them. The one performing better is the one that

Table 4 Collisions

	100	200	300	400	500	600	700
NS_Rad_DB	0.02 ±0.20	0.25 ±1.54	0.97 ±3.19	5.19 ±7.96	15.81 ±18.49	37.22 ±25.03	74.66 ±48.04
NS_Rad_EE	0.00 ±0.00	0.22 ±1.41	1.55 ±3.98	5.30 ±7.79	19.03 ±19.73	38.92 ±28.06	69.96 ±48.85
S_Rad_DB	0.09 ±0.31	1.01 ±4.31	16.60 ±25.11	101.98 ±85.64	391.55 ±183.51	810.55 ±279.51	1428.18 ±416.31
S_Rad_EE	0.08 ±0.56	1.02 ±4.02	23.54 ±32.88	128.26 ±96.66	367.17 ±147.08	812.67 ±289.21	1603.83 ±434.92
NS_Fix_DB	0.00 ±0.00	0.09 ±0.73	0.67 ±3.16	8.40 ±10.99	29.43 ±24.07	55.29 ±33.78	102.13 ±54.40
NS_Fix_EE	0.00 ±0.00	0.09 ±0.64	1.53 ±4.18	9.62 ±11.57	27.86 ±23.97	55.02 ±34.73	100.17 ±52.64
S_Fix_DB	0.04 ±0.40	2.37 ±8.33	19.18 ±25.23	116.10 ±85.05	379.34 ±150.99	853.52 ±225.14	1526.04 ±448.75
S_Fix_EE	0.04 ±0.40	2.67 ±7.59	16.85 ±23.33	116.72 ±90.84	421.88 ±179.17	932.97 ±275.46	1644.51 ±395.78
NS_Pow_DB	0.00 ±0.00	0.10 ±0.64	2.31 ±4.94	17.30 ±19.36	46.00 ±35.79	128.35 ±60.79	197.53 ±84.35
NS_Pow_EE	0.00 ±0.00	0.07 ±0.50	1.33 ±3.55	17.92 ±22.78	46.76 ±36.22	144.35 ±72.01	230.86 ±107.36
S_Pow_DB	0.12 ±0.86	1.13 ±4.31	13.72 ±20.76	121.49 ±96.05	408.75 ±190.54	943.61 ±318.06	1688.47 ±458.08
S_Pow_EE	0.14 ±1.00	1.64 ±5.25	17.43 ±28.63	151.34 ±113.40	426.24 ±211.19	960.23 ±318.92	1769.64 ±444.99

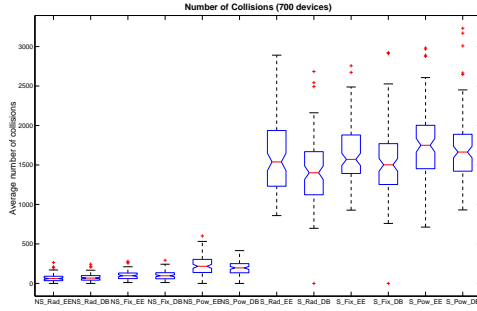


Figure 7 Collisions in the Broadcasting Process.

chooses randomly the waiting time from a fixed interval between $[0,1]$ (NS_Rad_EE and NS_Rad_DB). The second is the one taking a fixed delay inversely proportional to the distance from the source node (*powerDelay*). And the one that performs worse chooses randomly the waiting time from a fixed interval between $[0, powerDelay]$.

Obviously, if the waiting time is not stopped, the delay of the broadcasting process is increased but the number of collisions is highly reduced. We leave for future work a deep study on the compromise of both, reducing the number of collision and the dissemination delay.

7. CONCLUSIONS AND FUTURE WORK

In this paper we are proposing an energy saving strategy for the well known distance based broadcasting algorithm DB, and we are also studying some different parametrization for establishing the waiting time before the candidate nodes forward the broadcasting message.

For decreasing the energy consumption we propose to perform a reduction in the transmission power when possible. That is, when the fall in the transmission power does not imply the loss of any neighbor, and therefore does not provoke network partitions. This is really useful when the network is not very dense reducing up to 86.97% in the best case, but when the number of devices is big, the node does not reduce the transmission power so much since there are usually nodes close to the border. Anyway, in the worst case, this strategy of reducing the transmission power is saving at least 7.55% of energy per forwarded message.

A study of different strategies for establishing the settings of the delay that nodes perform when they receive a broadcasting message was made. The original DB sets a random delay within a fixed interval, and when a message is heard for the second time, it cancels the waiting time and decides to forward the message or not. In this work we are proposing different strategies: (1) Allowing the node to finish the waiting time when the same message is received twice. (2) The value of the delay is fixed (*powerDelay*), and inversely proportional to the distance from the source node. Finally (3), the delay is randomly chosen from an interval which size varies from 0 to *powerDelay*. After some experiments, we can state that allowing the node to finish the waiting time highly reduces the number of collisions in the network. This reduction is getting more importance as the network is becoming denser.

As a result from the experiments performed, we can say that NS_Rad_EE is the one that generally behaves better. In terms of the coverage achieved there is no statistical difference with the best one in any case (all proposals behaves similarly). Regarding the energy consumption is the best one with statistical differences in many cases or does not have any with the best. And finally in terms of the number of collisions is also the one that generally gets better results.

In this work we are presenting a protocol that is able to reduce energy without degrading the network connectivity and that also reduces the number of collisions in a 95.41%.

As future work, we plan to study a technique to highly reduce the energy consumption not only in sparse networks, but also in dense ones. We would also like to optimize the protocol and find out the thresholds that

maximize the coverage obtained by the broadcasting process, using the minimum energy consumption and number of collisions. And finally, we also want to find a tradeoff between the delay and the number of collisions.

REFERENCES

- [1] E. Alba and B. Dorronsoro, *CELLULAR GENETIC ALGORITHMS*, ser. Operations Research/Computer Science Interfaces. Springer-Verlag, 2008.
- [2] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *ACM/IEEE Int. Conf. on Mobile computing and networking*, 1999, pp. 151–162.
- [3] X. Li, T. D. Nguyen, and R. P. Martin, "Using adaptive range control to optimize 1-hop broadcast coverage in dense wireless networks," in *SenSys*, 2003, pp. 314–315.
- [4] J. Gomez and A. T. Campbell, "Variable-range transmission power control in wireless ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 1, 2007, pp. 87–99.
- [5] M. Cagalj, J.-P. Hubaux, and C. Enz, "Minimum-energy broadcast in all-wireless networks: Np-completeness and distribution issues," in *Int. Conf. on Mobile Computing and Networking*, 2002, pp. 172–182.
- [6] M. Cagalj, J.-P. Hubaux, and C. C. Enz, "Energy-efficient broadcasting in all-wireless networks," *Wirel. Net.*, vol. 11, no. 1, pp. 177–188, 2005.
- [7] W. Liang, R. Brent, Y. Xu, and Q. Wang, "Minimum-energy all-to-all multicasting in wireless ad hoc networks," *Trans. Wireless Comm.*, vol. 8, no. 11, 2009, pp. 5490–5499.
- [8] D. Liarokapis, A. Shahrabi, and A. Komninos, "Diba: An adaptive broadcasting scheme in mobile ad hoc networks," in *Communication Networks and Services Research*, 2009, pp. 224–231.
- [9] C. Zhi-yan, J. Zhen-zhou, and H. Ming-zeng, "An energy-aware broadcast scheme for directed diffusion in wireless sensor network," *Journal of Communication and Computer*, vol. 4, no. 5, 2007, pp. 28–35.
- [10] X. Chen, M. Faloutsos, and S. V. Krishnamurthy, "Power adaptive broadcasting with local information in ad hoc networks," in *Conference on Network Protocols*. IEEE Computer Society, 2003, p. 168.
- [11] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, *Mobile Ad Hoc Networking*. Wiley-IEEE Press, 2004.
- [12] V. Kawadia and P. Kumar, "Principles and protocols for power control in wireless ad hoc networks," *Selected Areas in Communications, IEEE Journal on*, vol. 23, no. 1, 2005, pp. 76–88.
- [13] C. Zhu, M. Lee, and T. Saadawi, "A border-aware broadcast scheme for wireless ad hoc network," in *Consumer Communications and Networking Conference. CCNC 2004. First IEEE*, 2004, pp. 134–139.
- [14] A. Benslimane, "Optimized dissemination of alarm messages in vehicular ad-hoc networks (VANET)," in *HSNMC*, 2004, pp. 655–666.
- [15] M. Lacage and T. R. Henderson, "Yet another network simulator," in *WNS2, workshop on ns-2: the IP network simulator*, 2006, p. 12.
- [16] "ns3 project website, <http://www.nsnam.org/>."
- [17] R. B. Groenevelt, E. Altman, and P. Nain, "Relaying in mobile ad hoc networks: The brownian motion mobility model," *Journal of Wireless Networks*, 2006, pp. 561–571.